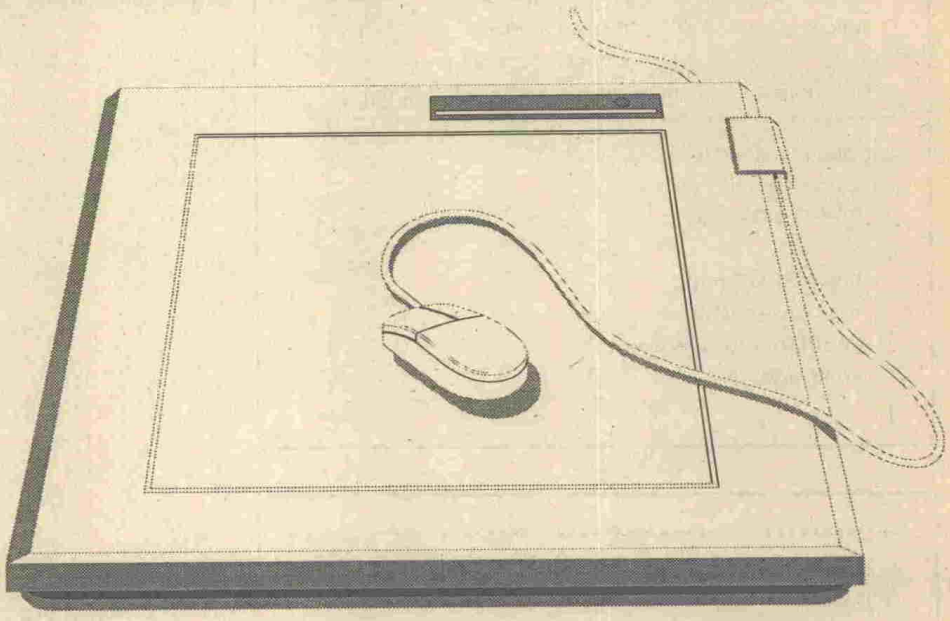
ภา สิงห์บุบผา
SOFT MAIL

สร้างฟังก์ชันเพื่ออินเตอร์เฟส เม้าส์ เข้ากับคลิปเปอร์

สร้างฟังก์ชันจากภาษาซี เพื่อเชื่อมโยงกับคลิปเปอร์ ให้ใช้งานเม้าส์ได้



คงปฏิเสธไม่ได้ว่าเม้าส์เป็นอุปกรณ์มาตรฐานสำหรับคอมพิวเตอร์ในปัจจุบัน อันมีผลมาจากโปรแกรมวินโดวส์ที่จำเป็นต้องใช้เม้าส์ประกอบ สำหรับโปรแกรมประยุกต์ที่รันบนระบบดอสในปัจจุบันส่วนใหญ่ก็สามารถเชื่อมโยงกับเม้าส์ได้ ในคลิปเปอร์ไม่มีฟังก์ชันสำหรับการติดต่อกับเม้าส์ หากว่าต้องการเขียนโปรแกรมเพื่อเชื่อมโยงกับเม้าส์อาจต้องใช้ไลบรารีอื่น ๆ มาช่วย เช่น Library for Clipper หรือไลบรารี VI (Visual Interface for Clipper 5.0) ซึ่งทำงานคล้าย ๆ วินโดวส์ คือจะเข้าสู่สภาวะกราฟิกโหมดและเชื่อมโยงกับเม้าส์ อีกวิธีหนึ่งที่จะทำให้คลิปเปอร์เชื่อมโยงกับเม้าส์ได้ คือ การสร้างฟังก์ชันจากภาษาอื่นแล้วนำมาเชื่อมโยงกับคลิปเปอร์ ในที่นี้ผู้เขียนจะใช้ภาษาซีในการเรียกอินเตอร์รัพต์

หากผู้อ่านได้ศึกษาบทความการเชื่อมโยงซีเข้ากับคลิปเปอร์ในสองตอนที่แล้วจะเข้าใจได้ง่ายขึ้น การทำงานกับเม้าส์จะเกี่ยวข้องกับเหตุการณ์ต่าง ๆ ดังนี้ การเช็คการติดตั้งเม้าส์ การกำหนดการแสดงหรือไม่แสดงเคอร์เซอร์ของเม้าส์ การกำหนดวินโดว์สำหรับเม้าส์ การรีเทอร์นค่าตำแหน่งเคอร์เซอร์เม้าส์และการกดปุ่มของเม้าส์ ในการติดต่อกับเม้าส์ผู้อ่านต้องทำการติดตั้งเม้าส์โดยใช้โปรแกรม MOUSE.COM สำหรับอินเตอร์รัพต์ที่ใช้ติดต่อกับเม้าส์เป็นอินเตอร์รัพต์ที่ 33H ซึ่งเป็นอินเตอร์รัพต์ของดอส

การติดต่อกับเม้าส์

โปรแกรมที่เขียนขึ้นมาเพื่อเรียกใช้เม้าส์ต้องติดต่อกับเม้าส์เพื่อรับรู้ว่ามีการติดตั้งเม้าส์ ซึ่งต้องเรียกใช้อินเตอร์รัพต์

```
/*************************************************
* Function Name.... mReset()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... mReset() -> lmemVar
* Return..... .T. = Mouse initialized
*                .F. = Mouse could not be init.
* Description..... check mouse initialized .OR. not
*************************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MRESET()
{
   union REGS regs;
   regs.x.ax=0x00;
   int86(0x33,&regs,&regs);
   _retl(regs.x.ax);
}
```

รูปที่ 1
โปรแกรม MRESET.C
ฟังก์ชัน mReset()

```
*************************************************
// Program name   : tmouse.prg
// Description    : check mouse initial or not
*************************************************
check := mReset()
IF check
      ? "Mouse initialized"
ELSE
      ? "Mouse not initialized"
ENDIF
RETURN
```

รูปที่ 2
โปรแกรม tmouse.prg

ที่ 33H ฟังก์ชันที่ 0 การเรียกใช้ในภาษาซีอยู่ในรูปที่ 1

ฟังก์ชัน mReset() จะกำหนดให้รีจิสเตอร์ AX = 0 และจะรีเทอร์นค่าทางรีจิสเตอร์ AX โดยหากติดต่อกับเม้าส์ได้สำเร็จจะให้ค่าจริง หากติดต่อกับเม้าส์ไม่สำเร็จจะให้ค่าเท็จ รูปแบบการเรียกใช้ฟังก์ชันคือ

```
mReset() -> memVar
```

เมื่อคอมพายล์โปรแกรมด้วยคำสั่ง

```
QCL /c /AL Zl /Olt /Gs mreset.c
```

จะได้ไฟล์ mreset.obj ซึ่งเราจะเก็บรวบรวมไว้เป็นไลบรารี MOUSE.LIB ด้วยคำสั่งนี้

```
LIB mouse +mreset;
```

จะได้ไฟล์ MOUSE.LIB ทดลองเรียกใช้ฟังก์ชันดังโปรแกรมในรูปที่ 2 ซึ่งเป็นโปรแกรมเช็คว่ามีการติดตั้งเม้าส์หรือไม่

โปรแกรมที่ 2 ให้คอมพายล์ด้วยคลิปเปอร์และลิงค์ด้วยคำสั่ง

```
LINK/NOE/SE:512 tmouse,,,
llibce+mouse
```

เพื่อเป็นการรวบรัดคำอธิบาย ผู้เขียนจะให้รายละเอียดของการเรียกใช้อินเตอร์รัพต์ที่ 33 โดยใช้ฟังก์ชันหรือเซอร์วิซที่หมายเลขต่าง ๆ เพราะการคอมพายล์ การลิงค์ และการรวบรวมเก็บเป็นไลบรารีไว้เรียกใช้งานผู้อ่านคงมีความเข้าใจดีแล้ว ในตอนท้ายจะเป็นตัวอย่างการเรียกใช้ฟังก์ชันของเม้าส์โดยรวม และการเขียนเมนูแบบแถบแสงเชื่อมโยงกับเม้าส์

การแสดงเคอร์เซอร์ของเม้าส์

ปกติเคอร์เซอร์ของเม้าส์จะไม่แสดงให้เห็นทางจอภาพ การทำให้เคอร์เซอร์ของเม้าส์ปรากฏบนจอภาพจะเรียกใช้อินเตอร์รัพต์ที่ 33H ฟังก์ชันที่ 1 ดังรูปที่ 3 ซึ่งการเรียกใช้ฟังก์ชันนี้จะทำให้เคอร์เซอร์เม้าส์ปรากฏ คำอธิบายดูที่คอมเมนต์คงจะเข้าใจ

การซ่อนเคอร์เซอร์ของเม้าส์

เมื่อกำหนดให้เม้าส์เคอร์เซอร์ปรากฏแล้ว หากเลิกการทำงานของโปรแกรม หรือบางขณะต้องการซ่อนเม้าส์เคอร์เซอร์ก็สามารถทำได้ โดยจะเรียกใช้อินเตอร์รัพต์ที่ 33H ฟังก์ชันที่ 2 ดังรูปที่ 4 ซึ่งการเรียกใช้ฟังก์ชันนี้จะทำให้ไม่เห็นเคอร์เซอร์เม้าส์

```
/*************************************
* Function Name.... mCurOn()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... mCurOn()
* Return..... no return
* Description..... show mouse cursor
*************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MCURON()
{
    union REGS regs;
    regs.x.ax=0x01;
    int86(0x33,&regs,&regs);
}
```

รูปที่ 3 โปรแกรม mcuron.c ฟังก์ชัน mCurOn()

```
/*********************************************
* Function Name.... mCurOff()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... mCurOff()
* Return..... no return
* Description..... hide mouse cursor
*********************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MCUROFF()
{
    union REGS regs;
    regs.x.ax=0x02;
    int86(0x33,&regs,&regs);
}
```

รูปที่ 4 โปรแกรม mcuroff.c ฟังก์ชัน mCurOff()

```
/*****************************************
* Function Name... Mgetx()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... column := mGetX()
* Return..... column at mouse cursor
*****************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MGETX()
{
    union REGS regs;
    regs.x.ax=0x03;
    int86(0x33,&regs,&regs);
    _retni(regs.x.dx/8);
}
```

รูปที่ 5
โปรแกรม mgetx.c
ฟังก์ชัน mGetx()

ตำแหน่งเคอร์เซอร์เม้าส์

การเช็คตำแหน่งของเม้าส์ว่าอยู่ตำแหน่งใดของจอภาพ เช็คโดยเรียกใช้อินเตอร์รัพต์ของดอสหมายเลข 0X33 ฟังก์ชันที่ 3 โดยกำหนดค่ารีจิสเตอร์ AX = 0X33 แล้วรีเทอร์นค่าจากรีจิสเตอร์ DX ออกมาเป็นตำแหน่งในแนวแกน X ซึ่งต้องหารด้วย 8 ดังโปรแกรมในรูปที่ 5 สำหรับตำแหน่งในแนวแกน Y จะรีเทอร์นค่าจากรีจิสเตอร์ CX ซึ่งต้องหารด้วย 8 เช่นเดียวกัน ดังโปรแกรมในรูปที่ 6 โดยค่าที่รีเทอร์นกลับไปสู่คลิปเปอร์ทั้งแกน X และแกน Y จะเป็นข้อมูลชนิด NUMERIC

กำหนดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์

หากไม่ได้กำหนดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์ เคอร์เซอร์เม้าส์สามารถเคลื่อนที่ไปได้เต็มจอภาพ ในบางกรณีเรามีความต้องการจำกัดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์ โดยต้องจำกัดขอบเขตทางด้านซ้ายและขวาสำหรับการเคลื่อนที่ในแนวนอน และกำหนดขอบเขตบนและล่างสำหรับแนวตั้ง การจำกัดขอบเขตการเคลื่อนที่ของเคอร์เซอร์เม้าส์ในแนวนอนจะเรียกใช้ฟังก์ชันที่ 7 ส่วนการจำกัดขอบเขตการเคลื่อนที่ของเม้าส์ตามแนวตั้งจะใช้ฟังก์ชันที่ 8 ให้ผู้อ่านดูรูปที่ 7 ซึ่งฟังก์ชันจะรับค่าพารามิเตอร์จากคลิปเปอร์ทั้งหมด 4 ตัว (_parni(1) ถึง _parni(4)) ไม่มีการรีเทอร์นค่ากลับสู่คลิปเปอร์

```
/*******************************************
* Function Name.... mGety()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... row := mGetY()
* Return..... row at mouse cursor
*******************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MGETY()
{
    union REGS regs;
    regs.x.ax=0x03;
    int86(0x33,&regs,&regs);
    _retni(regs.x.cx/8);
}
```

รูปที่ 6 โปรแกรม mgety.c ฟังก์ชัน mGety()

```
/*************************************************
* Function Name.... MSetWin()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... MSetWin(<nX0>, <nY0>, <nX1>, <nY1>)
* Return..... no return
* Description..... Restricts mouse cursor movement to
*                  a rectangular window.
*************************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MSETWIN()
{
    union REGS regs;
    regs.x.dx=8*(_parni(2));
    regs.x.cx=8*(_parni(4));
    regs.x.ax=0x07;
    int86(0x33,&regs,&regs);
    regs.x.dx=8*(_parni(1));
    regs.x.cx=8*(_parni(3));
    regs.x.ax=0x08;
    int86(0x33,&regs,&regs);
}
```

รูปที่ 7 โปรแกรม msetwin.c ฟังก์ชัน mSetWin()

```
/*************************************************
* Function Name.... mStatus()
* Programmer..... Bhao Singboobpha
* Date..... 16-04-37
* Syntax..... nButtonDown := MStatus()
* Return..... A numeric value indicating
*             which mouse button are pressed.
*             Value   Meaning
*
*               0     No button.
*               1     Left button.
*               2     Right button.
*               3     Both button.
* Description..... Returns a value indicating which
*             buttons, if any, are now down.
*             When both buttons are up, the function
*             returns zero.
*************************************************/
#include "extend.h"
#include "dos.h"
CLIPPER MSTATUS()
{
    union REGS regs;
    regs.x.ax=0x03;
    int86(0x33,&regs,&regs);
    _retni(0);
    switch(regs.x.bx)
    {
     case 1 : _retni(1); break;
     case 2 : _retni(2); break;
     case 3 : _retni(3); break;
    }
}
```

รูปที่ 8 โปรแกรม mstatus.c ฟังก์ชัน mStatus()

การอ่านค่าการกดปุ่มเม้าส์

เม้าส์ในที่นี้จะเป็นเม้าส์ที่ใช้ปุ่
ซ้ายและขวา ดังนั้นจะมีสภาวะการกดปุ่
ดังนี้คือ ไม่กด กดปุ่มซ้าย กดปุ่มขวา แล
กดทั้งสองปุ่ม การอ่านค่าการกดปุ่มขอ
เม้าส์ใช้อินเตอร์รัพต์หมายเลข 33
ฟังก์ชันหมายเลข 3 ซึ่งจะรีเทอร์นค่
จากรีจิสเตอร์ BX แล้วส่งค่าไปยังคลิป
เปอร์เป็นชนิดตัวเลขตั้งแต่ 0-3 สำหรั

ค่าที่ได้จากการกดปุ่มเม้าส์สภาวะใดให้ดูได้จากคอมเมนต์ในโปรแกรมรูปที่ 8

ตัวอย่างการเรียกใช้ฟังก์ชันในคลิปเปอร์

สำหรับฟังก์ชันต่าง ๆ ที่เสนอมานี้ ผู้เขียนคิดว่าเพียงพอแล้วสำหรับการควบคุมเม้าส์เข้ากับโปรแกรมที่ผู้อ่านเขียนขึ้นมาใช้งานร่วมกับเม้าส์ในเท็กซ์-โหมด เมื่อผู้อ่านคอมพายล์โปรแกรมแล้วให้เก็บรวบรวมไว้ในไลบรารี MOUSE.LIB เพื่อเรียกใช้ต่อไป ในรูปที่ 9 จะเป็นตัวอย่างโปรแกรมของคลิปเปอร์เพื่อเรียกใช้ฟังก์ชันที่เกี่ยวกับเม้าส์ โดยจะกำหนดขอบเขตการเคลื่อนที่ของเม้าส์ให้อยู่ในกรอบสี่เหลี่ยมที่กำหนดขึ้นมา พร้อมทั้งแสดงพิกัดของเม้าส์ที่จะเปลี่ยนไปเมื่อเคอร์เซอร์เม้าส์เคลื่อนที่ และเมื่อมีการกดปุ่มเม้าส์จะแสดงสถานะการกดว่าเป็นปุ่มซ้ายหรือขวาพร้อมทั้งเสียงสูงและต่ำ เมื่อต้องการออกจากโปรแกรมให้กดปุ่มเม้าส์พร้อมกันทั้งสองปุ่ม เมื่อคอมพายล์โปรแกรมด้วยคลิปเปอร์แล้วให้ลิงค์ด้วยคำสั่งนี้

```
LINK/NOE/SE:512
chkwin,,,llibce+mouse
```

แถบแสงและเม้าส์

ในคลิปเปอร์การเลือกหัวข้อเลือกจะใช้การเลื่อนแถบแสงเพื่อเลือกหัวข้อ ในโปรแกรมรูปที่ 10 จะแสดงตัวอย่างการเขียนโปรแกรมแสดงแถบแสงให้เชื่อมโยงกับเม้าส์ โดยจะมีลักษณะดังนี้

1. เมื่อนำเคอร์เซอร์เม้าส์ไปคลิ๊กที่หัวข้อใด แถบแสงจะสว่างที่หัวข้อนั้น

```
//*********************************************************
// Program      : chkwin.prg
// Compile      : CLIPPER chkbuton
// Link         : LINK/NOE/SE:512 chkwin,,,llibce+mouse
// Run          : chkwin
//*********************************************************
CLEAR
r := mReset()
mCuron()
@ 4, 19 TO 20,61
mSetwin(5, 20, 19, 60)
DO WHILE .t.
        nButtons := mStatus()
        IF  nButtons = 3
           @ 1,1 SAY "Pressed both buttons for exit."
           @ 24, 79
           EXIT
        ELSEIF nButtons = 1
           @ 1,1 SAY "Press left botton. "
           TONE(330,2)

        ELSEIF nButtons = 2
             @ 1,1 SAY "Press right botton."
             TONE(660,2)
        ELSE
             xpos := mGetX()
             ypos := mGetY()
             @ 2,1 SAY "ROW:"+SUBSTR(STR(mgetx()),LEN(STR(mGetX()))-1,2)+;
                    " COLUMN:"+SUBSTR(STR(mgety()),LEN(STR(mGetY()))-1,2)
             nxpos := xpos
             nypos := ypos
             DO WHILE  nxpos = xpos .AND. nypos = xpos
                    nxpos := mGetX()
                    nypos := mGetY()
             ENDDO
        ENDIF
ENDDO
TONE(330,2)
mCuroff()
RETURN
```

รูปที่ 9 โปรแกรม CHKWIN.PRG ตัวอย่างการเรียกใช้ฟังก์ชันเม้าส์

รูปที่ 10 โปรแกรม MENU.PRG ตัวอย่างแถบแสงและเม้าส์

```
//************************************************
// Program    : menu.prg
// Compile    : CLIPPER menu
// Link       : LINK/NOE/SE:512 menu,,,llibce+mouse
// Run        : menu
//************************************************
IF !MReset()
  ? "No mouse"
  QUIT
ENDIF
mmenu_init(5)
CLEAR
mprompt(10,10," Account  ")
mprompt(11,10," Report   ")
mprompt(12,10," Screen 1 ")
mprompt(13,10," Screen 2 ")
mprompt(14,10," Screen 3 ")
@ 20, 0
@ 16, 10 SAY "You select : "+LTRIM(STR(mmenu(1)))
RETURN
////////////////////////////////////////////////////
FUNCTION mmenu_init(pmax_prompts)
PUBLIC menu_row[pmax_prompts]
PUBLIC menu_col[pmax_prompts+1]
PUBLIC menu_txt[pmax_prompts+1]
PUBLIC num_prompts
PUBLIC max_prompts
        max_prompts := pmax_prompts
        num_prompts := 0
RETURN .T.
FUNCTION mprompt(row, col, txt)
num_prompts := num_prompts+1
IF num_prompts > max_prompts
        RETURN .F.
ENDIF
menu_row[num_prompts] := row
menu_col[num_prompts] := col
menu_txt[num_prompts] := txt
RETURN .T.
////////////////////////////////////////////////////
FUNCTION mmenu(start_pos)
LOCAL i, left_down, right_down, horiz, vert, start, double_clicked
LOCAL vert_click, hor_click, key, mess
LOCAL start_mess, interval
PUBLIC cur_pos
interval := 0.5
cur_pos := start_pos
SET CURSOR OFF
left_down  := .F.
right_down := .F.
horiz := 0
vert := 0

FOR i = 1 TO num_prompts
    @ menu_row[i], menu_col[i] SAY menu_txt[i]
NEXT
SET COLOR TO N/W
@ menu_row[cur_pos], menu_col[cur_pos] SAY menu_txt[cur_pos]
SET COLOR TO
MCurOn()
double_clicked := .F.
DO WHILE !double_clicked
     mouse_read(@left_down, @right_down, @horiz, @vert)
    IF left_down .OR. right_down
          i := 1
          found := .F.
        DO WHILE i <= num_prompts .AND. !found
              found := vert = menu_row[i] .AND.;
                    horiz >= menu_col[i] .AND.;
                    horiz < menu_col[i] + LEN(menu_txt[i])
              IF !found
                  i++
              ENDIF
          ENDDO
          IF found
              IF cur_pos != i
                  move_to(i)
              ENDIF
              vert_hit  := vert
              horiz_hit := horiz
              start := seconds()
              DO WHILE (left_down .OR. right_down);
                    .AND. seconds()-start < 0.5 .AND.;
                    horiz =horiz_hit .AND. vert = vert_hit
                    mouse_read(@left_down,@right_
                    down,@horiz, @vert)
              ENDDO
              DO WHILE !left_down .AND. !right_down;
                    .AND. seconds()-start < 0.5 ;
                    .AND.horiz = horiz_hit;
                    .AND. vert = vert_hit
                    mouse_read(@left_down, @right_
                    down,; @horiz, @vert)
              ENDDO
              DO WHILE (left_down .OR. right_down);
                    .AND. seconds()-start < 0.5 ;
                    .AND.horiz = horiz_hit .AND. vert = vert h
                    mouse_read(@left_down,@right_
                    down,@horiz, @vert)
              ENDDO
              double_clicked := seconds() - start < 0.5;
```

ปที่ 10 (ต่อ)

```
                                .AND. vert = vert_hit;
                                .AND. horiz = horiz_hit
            ENDIF
        ENDIF
    ENDDO
    MCurOff()
    RETURN cur_pos
    ////////////////////////////////////////////////////////

    FUNCTION move_to(new_pos)
    MCurOff()
    @ menu_row[cur_pos], menu_col[cur_pos] SAY menu_txt[cur_pos]
    cur_pos := new_pos
    SET COLOR TO N/W
    @ menu_row[cur_pos], menu_col[cur_pos] SAY menu_txt[cur_pos]
    SET COLOR TO
    MCurOn()
    RETURN NIL
    ////////////////////////////////////////////////////////

    FUNCTION mouse_read(left_pressed, right_pressed, horizontal, vertical)
            left_pressed  := MStatus() = 1
            right_pressed := MStatus() = 2
            horizontal    := MGety()
            vertical      := MGetx()
    RETURN .T.
```

■

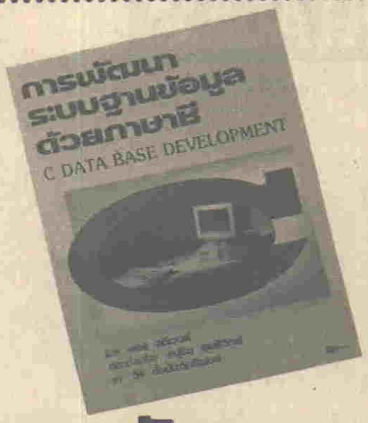
2. หากดับเบิ้ลคลิ๊กที่หัวข้อใด จะเป็นการเลือกหัวข้อนั้น

3. สามารถใช้เม้าส์กดปุ่มค้างไว้ที่แถบแสง แล้วลากแถบแสงไปยังหัวข้อที่ต้องการ

ให้ผู้อ่านพิจารณา user defined function แต่ละอัน เพื่อนำมาดัดแปลงใช้งาน ในการเข้าโปรแกรมครั้งแรกโปรแกรมจะเช็คว่าได้ติดตั้งเม้าส์หรือไม่ หากไม่จะออกจากโปรแกรมทันที หากติดตั้งเม้าส์จะเกิดหัวข้อให้ผู้อ่านทดลองใช้เม้าส์คลิ๊กหรือลากแถบแสง ผู้อ่านสามารถนำไปดัดแปลงทำเมนูเพื่อเลือกหัวข้อได้

สำหรับซอร์สโปรแกรมในฉบับนี้จะมีเยอะ ผู้เขียนได้จัดทำซอฟต์เมล์ชื่อ CMOUSE.EXE ซึ่งจะรวมฟังก์ชันภาษาซี โปรแกรมคลิปเปอร์และไลบรารี MOUSE.LIB ในตอนต่อไปจะเป็นเรื่องของ Visual Interface for Clipper หรือการอ่านภาพมาแสดง ท่านที่สงสัยสอบถามได้ที่ 036-222400 สวัสดีครับ

หนังสือดี ซีเอ็ด



การพัฒนาระบบฐานข้อมูลด้วยภาษาซี

■ *โดย ออล สตีเวนส์*
เรียบเรียงโดย อนุชิต สุมธิวิทย์/วีร์ ตั้งมั่นภักดีพงษ์
ราคาเล่มละ 150 บาท

หนังสือเล่มนี้ประกอบด้วยพื้นฐานและการออกแบบข้อมูลเบื้องต้น วิธีการสร้างระบบจัดการฐานข้อมูลอย่างง่ายด้วยภาษาซี พร้อมโปรแกรมที่สามารถนำไปใช้งานได้จริง ผู้อ่านจึงควรมีความสามารถในการเขียนโปรแกรมด้วยภาษาซีอยู่บ้างแล้ว นอกจากนี้จุดประสงค์ของหนังสือต้องการให้ผู้อ่านสามารถสร้าง tool เพื่อลดเวลาในการโปรแกรมลงได้อีกด้วย หนังสือเล่มนี้จึงเหมาะสำหรับผู้ที่สนใจเรื่องฐานข้อมูล และผู้ที่รักการเขียนโปรแกรมทั่วไป

หาซื้อได้ตามร้านหนังสือทั่วไป
หรือสั่งซื้อโดยตรงได้ที่

บริษัท ซีเอ็ดยูเคชั่น จำกัด
800/43-45 ซอยตระกูลสุข ถนนอโศก-ดินแดง ห้วยขวาง กรุงเทพฯ 10400 โทร. 248-6280-9 (เวลาทำการ จันทร์-ศุกร์ 8.10-17.30 น.) โดยส่งธนาณัติสั่งจ่าย ปท.ดินแดง ในนามบริษัทฯ หรือเช็คส่วนตัวของธนาคารในเขตกรุงเทพฯ พร้อมค่าจัดส่ง

ค่าจัดส่ง สั่งซื้อไม่เกิน 100 บาท ค่าจัดส่ง 8 บาท สั่งซื้อเกิน 100 บาท ค่าจัดส่ง 8%

พิเศษ สำหรับสมาชิกวารสารของซีเอ็ด สั่งซื้อโดยตรงที่บริษัทฯ ได้ในราคาลด 15% เพิ่มค่าลงทะเบียน 4 บาท ค่าจัดส่งฟรี โดยระบุหมายเลขสมาชิกด้วย